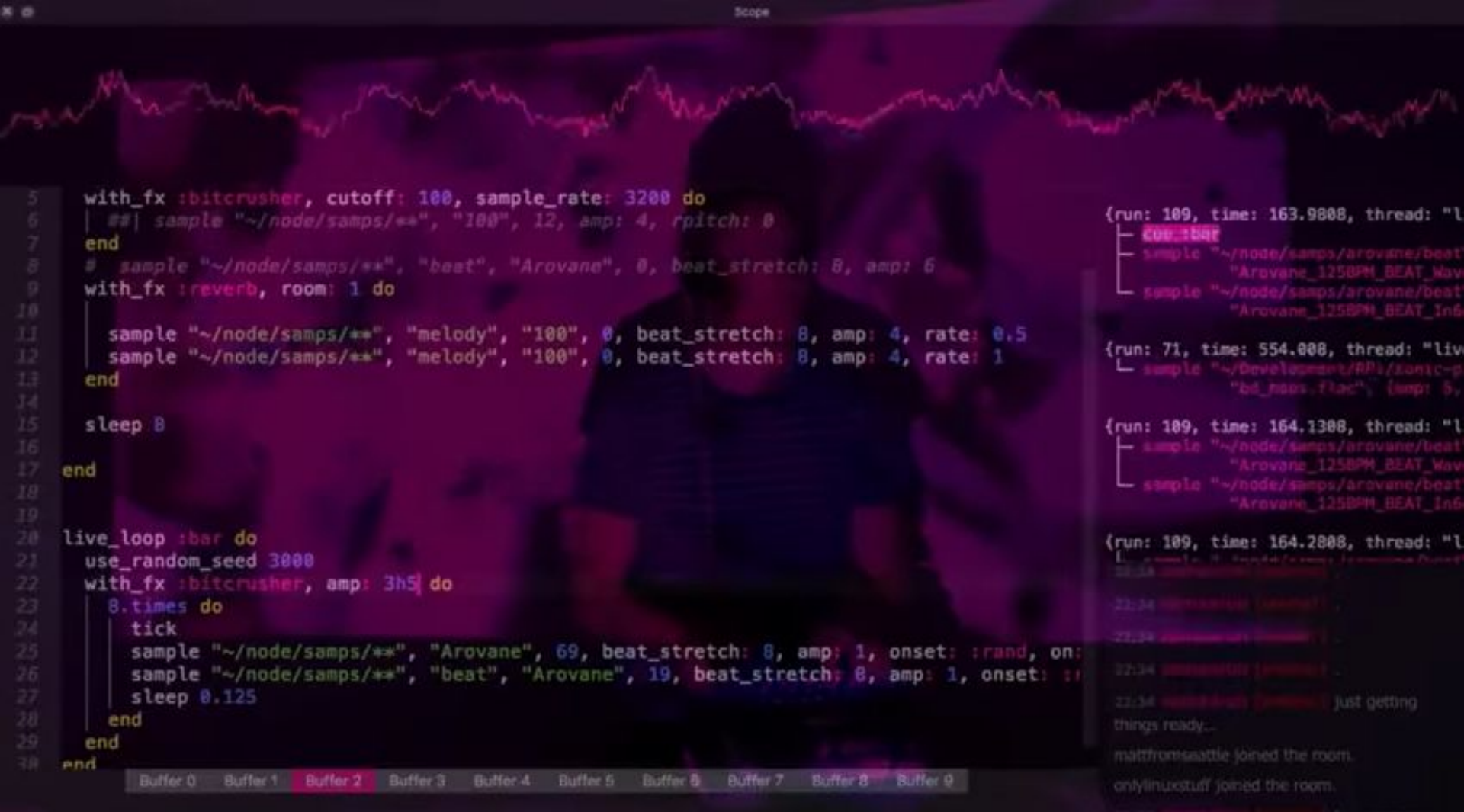


```
music_as :code  
code_as  :art
```

v2.11



[Sam Aaron live coding an ambient electro set w/ Sonic Pi](#)

[Sam Aaron - TEDx - Programming as Performance](#)

Inhalte

Sonic Pi's API in 10 Min

Kinderlied coden

Ein paar Gedanken zu live coding und Schule

Was ist sonst noch möglich?

Rocking mit Sonic Pi - ein Algorave(-versuch;))

Sonic Pi's API in 10 Min

Anweisungen

play 60

sample :bd_tek

sleep 1

Sonic Pi's API in 10 Min

Synths

use_synth :beep

play 60, release: 2, amp: 1

Sonic Pi's API in 10 Min

Samples

sample :loop_amen

use_sample_bpm :loop_amen

sample :loop_amen, rate: 0.5, cutoff: 90

sample :loop_amen, start: 0, finish: 0.5

Sonic Pi's API in 10 Min

```
# Effects  
with_fx :reverb do  
  irgendEinCode  
end
```

Sonic Pi's API in 10 Min

```
# Sequencing  
live_loop :demo do  
  irgendEinCode  
end
```


Informatische Grundkonzepte

2.times do

 play 70

 sleep 1

 x = 70

 play x + 2

 sleep 1

 play 74

 sleep 1

 play 70

 sleep 1

end

<- Schleife (mit end abschliessen)

<- Anweisung

<- Variable

Kinderlied coden

Mehrstimmigkeit

```
live_loop :1Stimme do
  2.times do
    irgendEinCode
  end
end
```

```
live_loop :2Stimme do
  einAndererCode
end
```

<- live_loop ist eine Endlosschleife und läuft in einem eigenständigen Thread (kann man sich als weiteren Computer vorstellen). Diese Threads können gesynct werden.

Download:

[Bruder Jakob.rtf](#)

Ein paar Gedanken zu live coding und Schule

- Grundkonzepte der Informatik sind mit Sonic Pi im Moment und intuitiv erfahrbar (auditive Rückmeldung)
- Sonic Pi bietet die Möglichkeit, den Computer als Musik-Instrument zu erfahren.
- Sonic Pi eignet sich nur begrenzt für Musiktheorie in der Volksschule.

Was ist sonst noch möglich?

```
test = "~/Dropbox/Tagung_19_11_16/bobbyFeeling.aiff"  
use_bpm 80  
bla = (0..1).step(1/16.0).each_cons(2).collect do |s, f|  
  {start: s, finish: f}  
end.ring
```

```
live_loop :blabla do  
  sample test, bla.tick, rpitch: (knit 0, 32, -5, 16, 0, 16).tick(:bob)  
  sleep 0.25  
end
```

```
live_loop :bd do  
  sample :bd_tek  
  sleep 1  
end
```

Was ist sonst noch möglich?

Samples können ganz einfach via Pfad angesprochen werden. Nur .wav und .aiff

```
test = "~/Dropbox/Tagung_19_11_16/bobbyfeeling.aiff"
```

```
use_bpm 80
```

```
bla = (0..1).step(1/16.0).each_cons(2).collect do |s, f|
```

```
  {start: s, finish: f}
```

```
end.ing
```

```
live_loop :blabla do
```

```
  sample test, bla.tick, rpitch: (knit 0,
```

```
  sleep 0.25
```

```
end
```

Ein Range wird in 16 Teile zerlegt und in der Variable bla gesammelt.

```
  tick(:bob)
```

Der Start- und Endpunkt des zu spielenden Sampleauszuges

```
live_loo
```

```
  sample
```

```
  sleep 1
```

```
end
```

Sonic Pi

Run ▶ Stop ■ Rec ● Save ♥ Load 📁 Size — Size + Scope 📡 Info ☆ Help 💬 Prefs 🔊

```
8
9
10 live_loop :blabla do
11   sample test, bla.tick, rp
12   sleep 0.25
13 end
14
15 live_loop :bd do
16   15.times do
```

Hilfestellungen und Tutorials sind in der "Help" Section in Sonic Pi bereits integriert.

botokoll

```
bobbyFeeling.aiff", {st
time: 0.9375, thread: "li
.blabla
sample "~/Dropbox/Tagung_19_11_
"bobbyFeeling.aiff", {
run: 85, time: 1.125, thread: "liv
cue :blabla
sample "~/Dropbox/Tagung_19_11_
"bobbyFeeling.aiff", {
run: 85, time: 1.3125, thread: "li
cue :blabla
sample "~/Dropbox/Tagung_19_11_
"bobbyFeeling.aiff", {
run: 85, time: 1.5, thread: "live_...
```

Puffer 0 Puffer 1 Puffer 2 Puffer 3 Puffer 4

pick
pitch_to_ratio
play
play_chord
play_pattern
play_pattern_timed
print
puts
quantise

Play cue

play note (synth, sample, or file) [options]

Play note with current synth or sample and options which include control of an amplitude envelope with **attack:**, **decay:**, **sustain:** and **release:** phases. These phases are triggered in order, so the duration of the sound is attack + decay + sustain + release times. The duration of the sound does not affect any other notes. Code continues executing whilst the sound is playing through its envelope phases.

If **duration:** is supplied and **sustain:** isn't, it causes **sustain:** to be set so that all four phases add up to the duration.

Tutorial Beispiele Synths Fx Samples **Sprache**

Sonic Pi v2.11 on Mac